

[0001] SYSTEM AND METHOD FOR REDUCING POWER CONSUMPTION BY ESTIMATING ENGINE LOAD AND REDUCING ENGINE CLOCK SPEED

[0002] CROSS REFERENCE TO RELATED APPLICATION

[0003] This application is a continuation of U.S. Patent Application Serial No. 09/767,086, filed January 22, 2001, which is incorporated by reference as if fully set forth.

[0004] BACKGROUND

[0005] The invention relates to power management in a computer system. More particularly, the invention relates to reducing power consumption of microprocessors in a computer system.

[0006] Reducing power consumption in computer systems is highly desirable. Reduced power consumption decreases the heat generated by the system. As the packaging of computer systems becomes more compact, the dissipation of heat generated by the system becomes problematic. Accordingly, it is desirable to reduce the generated heat by reducing power consumption. Additionally, portable or handheld computer systems usually rely on portable power supplies or batteries. Lower power consumption prolongs the usage of a single battery or power supply without a recharge.

[0007] The clock frequency of a microprocessor has a highly correlated relationship with the power consumption. A microprocessor running with a higher clock frequency consumes more power and produces more heat than the same microprocessor running with a lower clock frequency. A typical computer system fixes the clock frequency of its microprocessor at initialization. However, instead of running at the initialized clock frequency in both busy or idle modes, it is better to switch the system to a lower clock frequency if the computer is in an idle mode.

[0008] One approach to triggering a reduction in clock frequency is to monitor

inputs from input devices, such as a keyboard, mouse, or trackball. If there is no input from any of the input devices, for a predetermined period of time, the system will automatically switch from a normal system clock frequency to a slower one. The clock frequency returns to normal when the system receives an input from an input device, such as a keystroke or mouse movement. The input monitoring approach has drawbacks. Although the computer is not receiving an input, it may be engaged in heavy data processing, which a reduced clock frequency hinders this heavy processing.

[0009] Another approach for triggering a reduction in clock frequency is to analyze the number of instructions executed by the microprocessor for a predetermined period of time. If the number of executed instructions is low, the clock frequency is reduced. If the number of executed instructions increases, the clock frequency is increased to normal. One drawback to analyzing past instruction execution is that the past number of instructions may not accurately reflect future processing requirements. Accordingly, it is desirable to have alternate approaches to reduce power consumption.

[0010] SUMMARY

[0011] A computer system has a processor and a queue for storing instructions for execution by the processor. The processor is capable of being clocked at a plurality of different clock frequencies. A number of instructions in the queue is measured. A particular clock frequency is selected for the microprocessor based, in part, on the determined number of queued instructions.

[0012] BRIEF DESCRIPTION OF THE DRAWING(S)

[0013] Figure 1 is a simplified diagram illustrating the functional relationships among the microprocessor, system clock and instruction cache.

[0014] Figure 2 is a functional block diagram of a computer system with separate reducing power consumption mechanisms for the microprocessor and the graphics processor.

[0015] Figure 3 is a detailed block diagram of the clock estimation device.

[0016] Figure 4 is a functional block diagram of a computer system with long term and short term load estimation mechanisms.

[0017] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0018] A computer system for reducing power consumption mechanism with various clock frequencies is shown in Figure 1. The clock 55 provides a clocking signal for a microprocessor 50. The microprocessor 50 may be a central processing unit (CPU), a graphics processor, such as a D3D engine, or any other processor. The microprocessor 50 has an associated instruction cache 65. The cache 65 stores instructions for execution by the microprocessor 50. The cache 65 may be on the same chip as the microprocessor 50 or off chip. The system also has a memory buffer 66 storing instructions to be input to the instruction cache 65.

[0019] A load and estimation clock device 75 is used to control the clocking rate of the system. The estimation device 75 controls the frequency of the clock signal being input into the microprocessor 50. Based on a control signal from the estimation device 75, the clock 55 outputs the selected clocking frequency for the microprocessor among a plurality of available clock frequencies.

[0020] One approach to selecting the clock frequency is to analyze the queued instructions. Initially, the microprocessor 50 is set to run at an initialization clock frequency. All instructions waiting to be executed by the microprocessor 50 will be first queued at the memory buffer 66. The instructions queued in the memory buffer 66 are fed to the instruction cache 65 to be queued for execution by the microprocessor 50. Subsequent clock frequencies are estimated based on the number of queued instructions.

[0021] To further optimize the system's performance, the estimation device 75 analyzes the types of instructions queued. Certain instructions require more intense processing than others. To compensate for the varying intensities, each instruction is weighted based on its intensity.

[0022] Additionally, the estimation device 75 may use the measured microprocessor temperature, as determined by the temperature monitor 70, to maintain the computer system at an acceptable range of operating temperature. If the temperature of the microprocessor 50 is approaching an unacceptable level, the power consumption may need to be reduced to prevent circuit damage regardless of the backlog of instructions. To compensate for temperature, the clock estimation device 75 factors in the measured temperature.

[0023] Figure 2 illustrates a computer system having separate monitoring mechanisms for a microprocessor 90 and a graphics processor 115. The microprocessor 90 and the graphics processor 118 each have their own clock estimation device 100, 105, respectively. Each estimation device 100, 105 controls the clocking frequency of its processor 90, 115.

[0024] One advantage to the individual monitoring of the two processors 90, 115 is that synchronization between the two estimation devices 100, 105 can be achieved. One estimation device 100, 105 can use the other device's 105, 100 information to determine its clocking frequency. One situation where this synchronization may be desirable is when one processor 90 has a larger backlog of instructions than the other 115. Although the estimation device 105 for the smaller backlog processor 115 indicates a higher clocking speed, the clocking speed may be reduced to equalize the backlog between the microprocessors 90, 115. The equalization may improve the system's performance.

[0025] Another approach to estimating the clock frequency is to analyze the complexity of a set of n instructions 155, as shown in Figure 3. Within the clock estimation device 100 or 105, an instruction queue monitoring and data collection block 130 collects information concerning the n instructions 155 which are queued in either queues 65 or 66. Based on the system design and the clock estimation algorithm latency, the set of n instructions may not be the next n instructions for execution by the microprocessor 150 in the queues 65, 66. To illustrate, the following instructions are queued for execution:

..., I_{k+n+1} , I_{k+n} , I_{k+n-1} , ..., I_{k+1} , I_k , ..., I ;

[0026] I is the instruction ready for execution. I_k is the first queued instruction selected for analysis. According to I_x , the last instruction selected for analysis is I_{k+n} . The resulting latency in the queue is $k-i$. Based on the flexibility of the clock adjustment circuitry, the number of instructions n and algorithm latency $k-i$ can be chosen to perform clock optimization frequently or in longer periods. A load estimation and clock estimation block 135 takes the collected data and estimates the required microprocessor performance. Based on the load estimation, a clock frequency is selected. To reduce processing, instead of analyzing the entire instruction set, a moving average of the instruction's intensity can be taken. The moving average allows for a new load estimate every instruction cycle.

[0027] One approach to load and clock estimation uses a fuzzy logic controller. Using fuzzy logic, the n instructions are grouped into fuzzy sets, such as five sets. To illustrate, five fuzzy sets based on estimated engine load are very high load, high load, medium load, low load and very low load. The fuzzy controller outputs a fuzzy variable. After defuzzification of the fuzzy variable, the clock frequency is determined. Additional inputs may be added to the fuzzy controller, such as the current clock frequency, the processor's draw of current and the temperature. To more adaptively adjust the clock control algorithm, a neural network controller is used. The control algorithm is learned by the neural network controller.

[0028] To enhance performance, a short term and a long term estimate may be used, as shown in Figure 4. Built onto the microprocessor chip 160 is a short term estimation load estimation blocks 165, 170. The short term blocks 165, 170 analyze a set of instructions 190 stored in a cache on the chip. Since the analysis is performed on a silicon level, the analysis is performed frequently.

[0029] Long term load estimation blocks 205, 210 analyze a set of instructions 195 queued off-chip, such as in an off-chip cache or in memory. Based on the long term analysis, a long term optimum clock estimation block 175 determines a preferred long term clocking frequency. The update of the long term frequency may be performed at

the same or a lower rate then the short term analysis. On chip, an optimum clock estimation block determines the clock frequency based on the short term analysis and the preferred long term clocking frequency. Using this two-tier approach, short term performance can be adjusted at a fast rate.

* * *